Microsistem cu microprocesorul 8086

Universitatea Politehnica Timișoara

Facultatea de Automatică și Calculatoare

2024-2025

Proiectarea Microprocesoarelor

**Bîtea Răzvan**

***Cerințe și specificații***

**Structura:**

- unitate entral cu microprocesorul 8086

- 128 KB memorie EPROM, utilizând circuite 27C512

- 64 KB memorie SRAM, utilizând circuite 62256

- interfaţă entra, cu circuitul 8251, plasată în zona 04D0H – 04D2H sau 05D0H – 05D2H, în funcţie de poziţia microcomutatorului S1

- interfaţă paralelă, cu circuitul 8255, plasată în zona 0250H– 0256H sau 0A50H – 0A56H, în funcţie de poziţia microcomutatorului S2

- o minitastatură cu 9 contacte

- 10 led-uri

- un modul de afişare cu 7 segmente, cu 8 ranguri (se pot afişa maxim 8 caractere hexa simultan)

- un modul LCD, cu 2 linii a câte 16 caractere fiecare, cu o interfaţă la alegerea studentului

**Programarea microsistemului:**

- rutinele de programare ale circuitelor 8251 şi 8255

- rutinele de emisie/ recepţie caracter pe interfaţa entra

- rutina de emisie caracter pe interfaţă paralelă

- rutina de scanare a minitastaturii

- rutina de aprindere/ stingere a unui led

- rutina de afişare a unui caracter hexa pe un rang cu segmente

***Descrierea hardware***

***Unitatea entral***

**Componente:**

- microprocesorul 8086

- generator de tact 8284A

- circuite pentru amplificarea și demultiplexarea magistralelor de adrese şi date

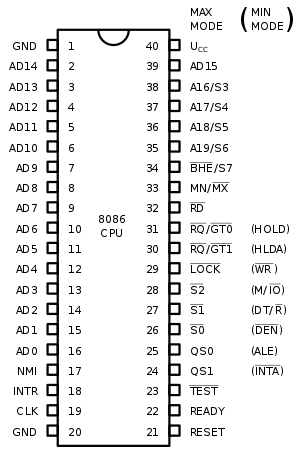
***Microprocesorul 8086***

Microprocesorul 8086 a apărut în anul 1978 și este primul și cel mai răspândit microprocessor pe 16 biți, care a cunoscut o largă utilizare. Apariția lui a fost urmată la scurt timp de o familie de componente : Intel 8284 (generator de tact), Intel 8288 (controler de magistrale) , Intel 8087 (coprocesor aritmetic), Intel 8089 (coprocesor de intrare-ieșire).

**Caracteristici:**

* Registrele interne și magistrala de date externă sunt de 16 biți
* Posibilitatea de a adresa direct 1 Mo de memorie
* Viteza mărită de lucru datorită atât frecvenței tactului cât și unei structuri interne bazată pe conceptul de suprapunere care permite aducerea în memorie, în avans, a instrucțiunilor, în timpul unor cicluri fără accese la magistrale
* Prezintă 2 moduri de lucru: modul minim, pentru aplicații relativ simple, precum și modul maxim, pentru aplicații mai complexe
* Magistrala de date și adrese sunt multiplexate iar o parte din terminalele de comandă au rol dublu; aceasta a permis încapsularea circuitului într-o capsulă de doar 40 terminale

**Modurile de lucru și terminalele:**



Existența a două moduri de lucru asigură separarea și protecția resurselor hardware și software între diferitele straturi de programare. Microprocesorul desi implementează această caracteristică, este cel mai rudimentar din acest punct de vedere.

Cele 2 moduri nu oferă priviligii diferite ci ele se recomandă în anumite configurații hardware: există terminalul MN/MX la care, prin 1 logic se cere modul minim, iar prin 0 logic cel maxim.

**Structura internă**

A diagram of a computer system

Description automatically generated

Microprocesorul 8086 cuprinde două

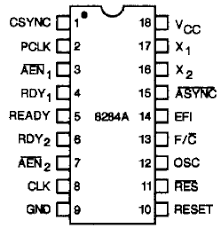
unități funcționale, care lucrează asincron și independent una față de cealaltă. Unitatea de execuție EU (Execution Unit) are rolul de a

executa instrucțiunile, iar unitatea de interfațare cu magistralele BIU (Bus Interface Unit) are rolul de a aduce instrucțiunile din memorie și de a transfera operanzii între unitatea de execuție și exteriorul

microprocesorului.

Existența acestora duce la creșterea vitezei de lucru datoriă suprapunerilor între activitățile lor.

***Generatorul de tact 8284A***



Circuitul are trei roluri într-o unitate centrală cu

microprocesorul 8086: generează tactul către microprocesor și pentru circuitele specializate pentru interfețe, generează semnalul READY către microprocesor, sincronizându-l cu tactul și generează semnalul de inițializare, RESET, către microprocesor,

sincronizându-l cu tactul.

***Circuitul registru 74x373***

A white grid with black text

Description automatically generatedCircuitul registru 74x373 este folosit pentru demultiplexarea liniilor de adresă. Este un registru cu ieșiri cu 3 stări, alcătuit din 8 bistabile. Are o intrare de validare pentru toate ieșirile /OC (dacă este pe 1 logic bistabilile trec în a 3-a stare) și o intrare pentru încărcarea bistabilelor G activă la 1 logic.

***Circuitul amplificator/ separator 74x245***

Circuitul 74x245 este folosit pentru amplificarea, respectiv separarea magistralelor bidirecționale, și este alcătuit din 8 perechi de porți cu 3 stări bidirecționale, cu o intrare de validare pentru toate porțile (/G) activă la 0 logic și o intrare de stabilire a direcției de transfer, DIR. Dacă DIR e 0 se citeşte, altfel se scrie.

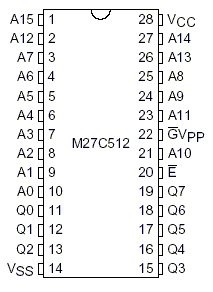
A white grid with black text

Description automatically generated

***Circuitele de memorie***

***EPROM***

Memoria EPROM (traducere liberă: memorie read only programabilă sau anulabilă) este un tip de memorie de calculator nevolatilă, adică o memorie care își păstrează datele chiaă și când i se întrerupe alimentarea cu curent electric. EPROM-urile sunt prin construcție o serie de tablouri de tranzistoare (porți logice) programate individual cu ajutorul unui dispozitiv electronic care furnizează tensiuni mai înalte decât cele pentru circuite digitale normale. Odată programate, un EPROM poate fi șters doar prin expunerea lui la lumină ultravioletă puternică. EPROM-urile au în partea superioară o fereastră de cuarț topit și resolidificat, prin care se vede cipul de siliciu propriu-zis. Acesta permite expunerea la luminp ultravioletă pentru ștergere (aducere la zero) a memoriei.

**Ci****rcuitul de memorie EPROM 27C512**

Capacitate: 64KB = 64 x 210 x 8 biți

Se vor folosi două circuite 27C512 pentru a obține o memorie EPROM de 128KB

***SRAM***

Memoria SRAM (Static Random Acces Memory) este un tip de memorie semiconductoare; static sublinează faptul că, spre deosebire de memoriile DRAM (Dynamic Random Acces Memory), nu mai este necesar un ciclu periodic de reîmprospătare (refresh). Memoriile SRAM folosesc circuite logice combinaționale pentru a memora fiecare bit. Datele depuse în memorie sunt stabile.

**Circuitul de memorie SRAM 62512**

Capacitate: 64KB = 64 x 210 x 8 biți

Se va folosi un singur circuit 62512.

***Decodificarea memoriei***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A19 | A18 | A17 | A16 | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| EPROM START | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EPROM END | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| SRAM START | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| SRAM END | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

SELECTIE EPROM = (~A19) & (~A18) & (~A17) => (~Y0)

SELECTIE SRAM = (~A19) & (~A18) & A17 & (~A16) => (~Y7)

Adresele de inceput si de sfarsit:

EPROM 128KB => 2^7 \* 2^10 = 2^17 biti

Spatiu de adresare: 00000H – 1FFFFH

SRAM 64KB => 2^6 \* 2^10 = 2^16 biti

Spatiu de stocare => 20000H – 2FFFFH

27C512 – 1 => EPROM

0000 0000 0000 0000 0000 (00000H)

0000 1111 111 1111 1111 (0FFFFH)

27C512 – 2 => EPROM

0001 0000 0000 0000 0000 (10000H)

0001 1111 1111 1111 1111 (1FFFFH)

62256 => SRAM

0010 0000 0000 0000 0000 (20000H)

0010 1111 1111 1111 1111 (2FFFFH)

***Interfaţa serială şi interfaţa paralelă***

***Interfața serială - Circuitul 8251***

Interfaţa serială este reprezentată de circuitele și programele de bază care asigură comunicare între unitatea centrală și echipamentele periferice, comunicare bit după bit.

Datorită costului mai redus și a rezistentei la perturbații, transferul de tip serial este util atunci când informația trebuie transmisă pe distanțe mari (peste 3 metri). Costul redus al transferului de tip serial este determinat de numărul firelor care leagă cele 2 echipamente, numărul mai mic de fire determinând un cost mai redus.

Circuitul 8251 (specializat pentru transferurile seriale) este un circuit USART (Universal Synchronous Asynchronous Receiver Transmitter) pentru comunicarea de tip serial.

Ca dispozitiv periferic, circuitul 8251 primește paralel informații de la CPU și le transmite serial, după conversie. De asemenea este capabil și sa primească date seriale din exterior şi să le transmită spre CPU paralel. De asemenea circuitul 8251 poate să lucreze în mod sincron şi asincron.

A diagram of a rectangular object with arrows

Description automatically generated Configurația Semnificația terminalelor

A white sheet with black text

Description automatically generated

***Interfața paralelă - Circuitul 8255***

Spre deosebire de transferul serial, la care transferul datelor se face bit după bit, la transferul paralel se transferă 8 biți simultan iar transferul este însoțit şi de semnale de dialog. Interfața paralelă se realizează cu ajutorul circuitului 8255.

A diagram of a rectangular object with arrows

Description automatically generatedComunicarea cu magistrala de date a unității centrale de prelucrare se face prin intermediul bufferului magistralei de date pe liniile D0 - D7. Prin acestea se transmit atât date, cât și cuvinte de control sau stare.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **/CS** | **/RD** | **/WR** | **A1** | **A0** | **Opera**ț**ia** |
| 0 | 1 | 0 | 0 | 0 | Scriere în portul A |
| 0 | 1 | 0 | 0 | 1 | Scriere în portul B |
| 0 | 1 | 0 | 1 | 0 | Scriere în portul C |
| 0 | 1 | 0 | 1 | 1 | Scriere în portul cuvântului de comandă |
| 0 | 0 | 1 | 0 | 0 | Citire din portul A |
| 0 | 0 | 1 | 0 | 1 | Citire din portul B |
| 0 | 0 | 1 | 1 | 0 | Citire din portul C |
| 0 | 0 | 1 | 1 | 1 | Fără operație – magistrala de date este în a 3-a stare |
| 0 | 1 | 1 | X | X | Fără operație – magistrala de date este în a 3-a stare |
| 1 | X | X | X | X | Magistrala de date este în a 3-a stare |

***Circuitul temporizator 8253A***

A diagram of a rectangular object with arrows

Description automatically generatedCircuitul 8253 (contor temporizator) generează rata de transfer pentru interfaţa serială, fiind conectat la terminalele TxC şi RxC ale 8251. Semnalul de selecţie al acestui circuit este de asemenea o ieşire a decodificatorului de porturi, având adresa 0C50H.

***Decodificarea porturilor***

**Interfata paralela**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ADR. | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | S2 |
| 0250H | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0256H | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0A50H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0A56H | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |

SELECTIE PAR1 = (~A10) & (~A8) & (~A7)

SELECTIE PAR2 = (~A10) & (~A8) & (~A7)

**Interfata seriala**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ADR. | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 | S1 |
| 04D0H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 04D2H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 05D0H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 05D2H | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |

SELECTIE SER1 = A10 & (~A8) & A7

SELECTIE SER2 = A10 & A8 & A7

**Interfaţa cu utilizatorul**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ADR. | A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | A7 | A6 | A5 | A4 | A3 | A2 | A1 | A0 |
| **Afișaje** | 0600H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0610H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0620H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0630H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0640H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0650H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0660H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0670H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| **LED** | 0680H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0690H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| **Tastatură** | 06A0H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 06B0H | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

0600H: 𝑆𝐸𝐿𝑆𝐴1= (~𝐴6) & (~𝐴5) & (~𝐴4) (Y0)

0610H: 𝑆𝐸𝐿𝑆𝐴2= (~𝐴6) & (~𝐴5) & A4 (Y1)

0620H: 𝑆𝐸𝐿𝑆𝐴3= (~𝐴6) & A5 & (~𝐴4) (Y2)

0630H: 𝑆𝐸𝐿𝑆𝐴4= (~𝐴6) & A5 & A4(Y3)

0640H: 𝑆𝐸𝐿𝑆𝐴5= 𝐴6 & (~𝐴5) & (~𝐴4) (Y4)

0650H: 𝑆𝐸𝐿𝑆𝐴6= 𝐴6 & (~𝐴5) & 𝐴4 (Y5)

0660H: 𝑆𝐸𝐿𝑆𝐴7= 𝐴6 & 𝐴5 & (~𝐴4) (Y6)

0670H: 𝑆𝐸𝐿𝑆𝐴8= 𝐴6 & 𝐴5 & 𝐴4 (Y7)

0680H: 𝑆𝐸𝐿𝑆𝐿1= (~𝐴6) & (~𝐴5) & (~𝐴4) (Y0)

0690H: 𝑆𝐸𝐿𝑆𝐿2= (~𝐴6) & (~𝐴5) & A4 (Y1)

06A0H: 𝑆𝐸𝐿ST1= (~𝐴6) & A5 & (~𝐴4) (Y2)

06B0H: 𝑆𝐸𝐿𝑆𝑇2= (~𝐴6) & A5 & A4 (Y3)

***Descrierea SOFTWARE***

;;rutina de programare a circuitului 8251 - cazul S1 = 0

;;adresele de port:

;;04D0H – comenzi/star

;;04D2H – date

MOV AL, 0CEH ;cuvânt de mod asincron

OUT 04D0H, AL ;adresa portului de stare/comanda

MOV AL, 15H ;cuvânt de comandă

OUT 04D0H, AL

RET

;;rutina de programare a circuitului 8251 - cazul S1 = 1

;;adresele de port:

;;05D0H – comenzi/stări

;;05D2H – date

MOV AL, 0CEH ;cuvânt de mod asincron

OUT 05D0H, AL ;adresa portului de stare/comanda

MOV AL, 15H ;cuvânt de comandă

OUT 05D0H, AL

RET

;;rutina de transmisie caracter pe interfaţa serială - caz S1 = 0

;;se transmite un caracter din CL

TR:

IN AL, 04D0H ;citire si testare rang TxRDY din cuv. de stare

RCR AL, 1 ;rotire dreapta cu carry

JNC TR ;pentru 0, reluam operaţia

MOV AL, CL ;se preia data din registrul CL

OUT 04D2H, AL ;trans. caracterul aflat initial in CL

RET ;pe portul de date

;;rutina de transmisie caracter pe interfaţa serială - caz S1 = 1

;;se transmite un caracter din CL

TR:

IN AL, 05D0H ;citire si testare rang TxRDY din cuv. de stare

RCR AL, 1 ;rotire dreapta cu carry

JNC TR ;pentru 0, reluam operaţia

MOV AL, CL ;se preia data din registrul CL

OUT 05D2H, AL ;trans. caracterul aflat initial in CL

RET ;pe portul de date

;;rutina de recepţie caracter pe interfaţa serială - cazul S1 = 0

;;se recepţionează caracterul in CL

REC:

IN AL, 04D0H ;citire si testare rang RxRDY din cuv. de stare

RCR AL, 2 ;verificăm bit-ul 1 (RXRDY)

JNC REC ;pentru 0 logic reluăm operaţia

IN AL, 04D2H ;preia data de la 8251

MOV CL, AL ;se depune data in registrul CL

RET

;;rutina de recepţie caracter pe interfaţa serială - cazul S1 = 1

;;se recepţionează caracterul in CL

REC:

IN AL, 05D0H ;citire si testare rang RxRDY din cuv. de stare

RCR AL, 2 ;verificăm bit-ul 1 (RXRDY)

JNC REC ;pentru 0 logic reluăm operaţia

IN AL, 05D2H ;preia data de la 8251

MOV CL, AL ;se depune data in registrul CL

RET

;;rutina de programare a interfeţei paralele - cazul S2 = 0

MOV DX, 0250H; adresa cuvântului de comandă

MOV AL, 81H; cuvânt de comanda:1000 0001

OUT DX, AL; port RCC

RET

;;rutina de programare a interfeţei paralele - - cazul S2 = 1

MOV DX, 0A50H; adresa cuvântului de comandă

MOV AL, 81H; cuvânt de comanda:1000 0001

OUT DX, AL; port RCC

RET

;;rutina de emisie caracter pe interfaţa paralelă - cazul S2 = 0

PAR:

IN AL, DX; citire și testare BUSY

RCR AL, 1 ;pentru a vedea daca receptorul este liber

JNC PAR

MOV AL, CL; se preia caracterul din registrul CL

MOV DX, 0250H

OUT DX, AL; transmit caracterul pe portul A

OR AL, 01H; trimit un semnal de strobe pe portul B

MOV DX, 0252H

OUT DX, AL; transmit caracterul modificat pe portul B

AND AL, 00H

OUT DX, AL; spun receptorului ca sunt date de citit

OR AL,01H

OUT DX, A; spun receptorului ca nu mai sunt date de transmis

RET

;;rutina de emisie caracter pe interfaţa paralelă - cazul S2 = 1

PAR:

IN AL, DX; citire și testare BUSY

RCR AL, 1;pentru a vedea daca receptorul este liber

JNC PAR

MOV AL, CL; se preia caracterul din registrul CL

MOV DX, 0A50H

OUT DX, AL; transmit caracterul pe portul A

OR AL, 01H; trimit un semnal de strobe pe portul B

MOV DX, 0A52H

OUT DX, AL; transmit caracterul modificat pe portul B

AND AL, 00H

OUT DX, AL; spun receptorului ca sunt date de citit

OR AL,01H

OUT DX, AL; spun receptorului ca nu mai sunt date de transmis

RET

**Rutina de aprindere/stingere a unui LED**

|  |  |  |
| --- | --- | --- |
| **LED1** | 11111110 | **FEH** |
| **LED2** | 11111101 | **FDH** |
| **LED3** | 11111011 | **FBH** |
| **LED4** | 11110111 | **F7H** |
| **LED5** | 11101111 | **EFH** |
| **LED6** | 11011111 | **DFH** |
| **LED7** | 10111111 | **BFH** |
| **LED8** | 01111111 | **7FH** |

;;rutina de aprindere/stingere a unui led pentru primele 8

;;anod comun : ledul va fi aprins când „0” este activ

MOV DX, 0680H

MOV AL, XX; 0680 setează semnalul SL1; XX selectează un anumit led

OUT AL

RET

;;rutina de aprindere/stingere a unui led pentru urmatoarele 2 led-uri

;;anod comun : ledul va fi aprins când „0” este activ

MOV DX, 0690H

MOV AL, XX; 0690 setează semnalul SL2; XX selectează un anumit led

OUT AL

RET

**Rutina de afișare a unui caracter HEXA pe un rang cu segmente**

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 00H | 11000000 | C0H |
| 1 | 01H | 11111001 | F9H |
| 2 | 02H | 10100100 | A4H |
| 3 | 03H | 10110000 | B0H |
| 4 | 04H | 10011001 | 99H |
| 5 | 05H | 10010010 | 92H |
| 6 | 06H | 10000010 | 82H |
| 7 | 07H | 11111000 | F8H |
| 8 | 08H | 10000000 | 80H |
| 9 | 09H | 10010000 | 90H |
| A | 0AH | 10001000 | 88H |
| B | 0BH | 10000011 | 83H |
| C | 0CH | 11000110 | C6H |
| D | 0DH | 10100001 | A1H |
| E | 0EH | 10000110 | 86H |
| F | 0FH | 10001110 | 8EH |

;;rutina de afişare a unui caracter hexazecimal, pe un rang cu 7 segmente

;;CL caracterul ce va fi afişat

;;CH rangul

SELNR:

CMP CL,NNH

; se determină cifra hexa ce trebuie afișată (NN se înlocuiește cu una din valorile : din tabel, iar SS se va alege corespunzzător lui NN)

JZ SELRG ; se sare la secvența de cod aferentă afișării lui NN concret

SELRG:

MOV AL,SSH; se încarcă valoarea scrisă în modul pentru afișarea cifrei

CMP CH,RH; se selectează rangul pe care se afișează cifra, înlocuind R

JZ PRT; se sare la secvența de cod aferentă lui R concret

PRT:

MOV DX,AAAAH; AAAA se înluciește în pe baza lui R concret

OUT DX,AL; se scrie valoarea SS la adresa AAAA

RET

;;rutina de programare a minitastaturii

RST:

MOV AL,FEH ;selectez prima coloana

OUT 06A0H,AL ;selectez ST1 (iesirea tastaturii)

IN AL, 06B0H ; ST2

;verific ce tasta am apasat

AND AL,01H ;verific prima pozitie

JZ TASTA0 ;salt la tasta 0 daca rezultatul este 0

IN AL, 06B0H; ST2

AND AL,02H ;verific a doua pozitie

JZ TASTA3 ;salt la tasta 3 rezultatul este 0

IN AL, 06B0H; ST2

AND AL,04H ;verific a treia pozitie

JZ TASTA6

MOV AL,FDH ;selectez a doua coloana

OUT 06A0H,AL ;selectez ST1 (iesirea tastaturii)

IN AL, 06B0H; ST2

;verific ce tasta am apasat

AND AL,01H ;verific prima pozitie

JZ TASTA1 ;salt la tasta 1 daca rezultatul este 0

IN AL, 06B0H; ST2

AND AL,02H ;verific a doua pozitie

JZ TASTA4 ;salt la tasta 4 rezultatul este 0

IN AL, 06B0H; ST2

AND AL,04H ;verific a treia pozitie

JZ TASTA7

MOV AL,FBH ;selectez a treia coloana

OUT 06A0H,AL ;selectez ST1 (iesirea tastaturii)

IN AL, 06B0H ; ST2

;verific ce tasta am apasat

AND AL,01H ;verific prima pozitie

JZ TASTA2 ;salt la tasta 2 daca rezultatul este 0

IN AL, 06B0H; ST2

AND AL,02H ;verific a doua pozitie

JZ TASTA5 ;salt la tasta 5 rezultatul este 0

IN AL, 06B0H ; ST2

AND AL,04H ;verific a treia pozitie

JZ TASTA8

;; blocul de mai jos se particularizeaza pentru diferite valori ale lui X

;; tratarea acţionării tastei X

TASTAX:

CALL DELAY ; se aşteaptă stabilizarea contactelor

WAITX:

IN AL, 06B0H; se citeşte din nou linia şi se asteapta dezactivarea tastei

AND AL, LLH; unde LLH este linia aferenta tastei X

JZ WAITX

CALL DELAY;

MOV CL,TTH; unde TTH este valoarea tastei

RET;reintoarcere din procedura

;;Linii:

;; prima linie - 01H = 0000 0001

;; a doua linie - 02H = 0100 0010

;; a treia linie - 04H = 0010 0100

;;Coloane

;; prima coloană – FEH = 1111 1110

;; a doua coloană – FDH = 1111 1101

;; a treia coloană – FBH = 1111 1011

***Scheme Hardware***

**Unitatea Centrală**

***A diagram of a circuit board

Description automatically generated***

**Conectarea memoriilor**

***A diagram of a computer circuit

Description automatically generated***

**Interfața serială și paralelă**

**A diagram of a circuit board

Description automatically generated**

**Decodificatorul de porturi**

**A diagram of a circuit

Description automatically generated**

**LED-uriA diagram of a circuit

Description automatically generated**

**MinitastaturaA diagram of a computer

Description automatically generated**

**Afișare segmente**

***A diagram of a circuit board

Description automatically generated***

***BIBLIOGRAFIE***

* Mircea Popa, Sisteme cu microprocesoare, Editura Orizonturi Universitare, Timișoara, 2003
* Mircea Popa, Proiectarea microsistemelor digitale, Editura Orizonturi Universitare, Timișoara, 2003
* Mircea Popa, Curs-PM, <https://cv.upt.ro/course/view.php?id=1590>, 2024